Weighted Model Counting Without Parameter Variables

Paulius Dilkas Vaishak Belle

University of Edinburgh, Edinburgh, UK

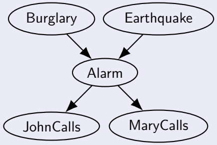
SAT 2021

The Computational Problem of Probabilistic Inference

ProbLog

```
0.001 :: burglary.
0.002 :: earthquake.
0.95  :: alarm     :- burglary, earthquake.
0.94  :: alarm     :- burglary, \+ earthquake.
0.29  :: alarm     :- \+ burglary, earthquake.
0.001 :: alarm     :- \+ burglary, \+ earthquake.
0.9   :: johnCalls :- alarm.
0.05  :: johnCalls :- \+ alarm.
0.7   :: maryCalls :- alarm.
0.01  :: maryCalls :- \+ alarm.
```

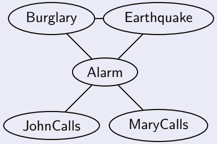
Bayesian Network



BLOG

```
random Boolean Burglary ~ BooleanDistrib(0.001);
random Boolean Earthquake ~ BooleanDistrib(0.002);
random Boolean Alarm ~
  if Burglary then
    if Earthquake then BooleanDistrib(0.95)
    else BooleanDistrib(0.94)
  else
    if Earthquake then BooleanDistrib(0.29)
    else BooleanDistrib(0.001);
random Boolean JohnCalls ~
  if Alarm then BooleanDistrib(0.9)
  else BooleanDistrib(0.05);
random Boolean MaryCalls ~
  if Alarm then BooleanDistrib(0.7)
  else BooleanDistrib(0.01);
```

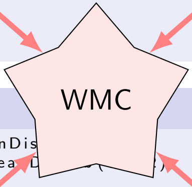
Markov Random Field



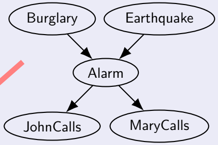
The Computational Problem of Probabilistic Inference

ProbLog

```
0.001  ::  burglary.
0.002  ::  earthquake.
0.95   ::  alarm     :- burglary, earthquake.
0.94   ::  alarm     :- burglary, \+ earthquake.
0.29   ::  alarm     :- \+ burglary, earthquake.
0.001  ::  alarm     :- \+ burglary, \+ earthquake.
0.9    ::  johnCalls :- alarm.
0.05   ::  johnCalls :- \+ alarm.
0.7    ::  maryCalls :- alarm.
0.01   ::  maryCalls :- \+ alarm.
```



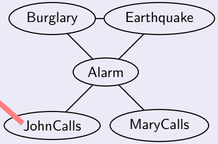
Bayesian Network



BLOG

```
random Boolean Burglary ~ BooleanDis
random Boolean Earthquake ~ Boolea  D      ib (   2)
random Boolean Alarm ~
  if Burglary then
    if Earthquake then BooleanDistrib(0.95)
    else BooleanDistrib(0.94)
  else
    if Earthquake then BooleanDistrib(0.29)
    else BooleanDistrib(0.001);
random Boolean JohnCalls ~
  if Alarm then BooleanDistrib(0.9)
  else BooleanDistrib(0.05);
random Boolean MaryCalls ~
  if Alarm then BooleanDistrib(0.7)
  else BooleanDistrib(0.01);
```

Markov Random Field



Weighted Model Counting (WMC)

- Generalises propositional model counting (#SAT)
- Applications:
  - graphical models
  - probabilistic programming
  - neural-symbolic artificial intelligence
- Main types of algorithms:
  - using knowledge compilation
  - using a SAT solver
  - manipulating pseudo-Boolean functions

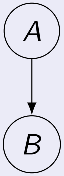
Example

$w(x) = 0.3,\ w(\neg x) = 0.7,$
$w(y) = 0.2,\ w(\neg y) = 0.8$

$\mathrm{WMC}(x \vee y) = w(x)w(y) + w(x)w(\neg y) + w(\neg x)w(y) = 0.44$

The Problem with Assigning Weights to Literals

A Simple Bayesian Network



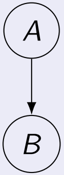
- from 2 binary variables
- to 8 variables and 17 clauses
- with lots of redundancy

Its WMC Encoding

```
p cnf 8 17
-2 -1 0
1 2 0
-3 1 0
-1 3 0
-5 -1 0
-5 -4 0
1 4 5 0
-6 -1 0
-6 4 0
-4 1 6 0
-7 1 0
-7 -4 0
-1 4 7 0
-8 1 0
-8 4 0
-4 -1 8 0
-4 0
c weights 1.0 1.0 0.5 1.0 \
0.5 1.0 1.0 1.0 0.6 1.0 \
0.4 1.0 0.1 1.0 0.9 1.0
```

The Problem with Assigning Weights to Literals

A Simple Bayesian Network



- from 2 binary variables
- to 8 variables and 17 clauses
- with lots of redundancy

Its WMC Encoding

```
p cnf 8 17
-2 -1 0
1 2 0
-3 1 0
-1 3 0
-5 -1 0
-5 -4 0
1 4 5 0
-6 -1 0
-6 4 0
-4 1 6 0
-7 1 0
-7 -4 0
-1 4 7 0
-8 1 0
-8 4 0
-4 -1 8 0
-4 0
c weights 1.0 1.0 0.5 1.0 \
0.5 1.0 1.0 1.0 0.6 1.0 \
0.4 1.0 0.1 1.0 0.9 1.0
```

$\neg x_1 \Leftrightarrow x_2$

$x_1 \Leftrightarrow x_3$

$\neg x_1 \land \neg x_4 \Leftrightarrow x_5$

$\neg x_1 \land x_4 \Leftrightarrow x_6$

$x_1 \land \neg x_4 \Leftrightarrow x_7$

$x_1 \land x_4 \Leftrightarrow x_8$

$\neg x_4$

Outline

Outline

WMC, Formally

Definition

A WMC instance is a tuple (ϕ, X_I, X_P, w), where

- X_I is the set of indicator variables,
- X_P is the set of parameter variables (with $X_I \cap X_P = \emptyset$),
- ϕ is a propositional formula in CNF over $X_I \cup X_P$,
- $w \colon X_I \cup X_P \cup \{\neg x \mid x \in X_I \cup X_P\} \to \mathbb{R}$ is a weight function
  - such that $w(x) = w(\neg x) = 1$ for all $x \in X_I$.

A More Expressive Alternative

Definition (Pseudo-Boolean Projection (PBP))

A PBP instance is a tuple (F, X, ω), where X is the set of variables, F is a set of two-valued pseudo-Boolean functions $2^X \to \mathbb{R}$, and $\omega \in \mathbb{R}$ is the scaling factor.

For any propositional formula ϕ over a set of variables X and $p, q \in \mathbb{R}$, let $[\phi]^p_q \colon 2^X \to \mathbb{R}$ be the pseudo-Boolean function defined as

$$[\phi]^p_q(Y) \coloneqq \begin{cases} p & \text{if } Y \models \phi \\ q & \text{otherwise} \end{cases}$$

for any $Y \subseteq X$.

From WMC to PBP

Example

- Indicator variable: x
- Parameter variables: p, q
- Weights: $w(p) = 0.2$, $w(q) = 0.8$, and $w(\neg p) = w(\neg q) = 1$

WMC Clause
$\neg x \Rightarrow p$
$p \Rightarrow \neg x$
$x \Rightarrow q$
$q \Rightarrow x$
$\neg x$

From WMC to PBP

Example

- Indicator variable: x
- Parameter variables: p, q
- Weights: $w(p) = 0.2$, $w(q) = 0.8$, and $w(\neg p) = w(\neg q) = 1$

WMC Clause	In CNF
$\neg x \Rightarrow p$	$x \lor p$
$p \Rightarrow \neg x$	$\neg x \lor \neg p$
$x \Rightarrow q$	$\neg x \lor q$
$q \Rightarrow x$	$x \lor \neg q$
$\neg x$	$\neg x$

From WMC to PBP

Example

- Indicator variable: x
- Parameter variables: p, q
- Weights: $w(p) = 0.2$, $w(q) = 0.8$, and $w(\neg p) = w(\neg q) = 1$

WMC Clause	In CNF	Pseudo-Boolean Function
$\neg x \Rightarrow p$	$x \vee p$	$[\neg x]_1^{0.2}$
$p \Rightarrow \neg x$	$\neg x \vee \neg p$	
$x \Rightarrow q$	$\neg x \vee q$	$[x]_1^{0.8}$
$q \Rightarrow x$	$x \vee \neg q$	
$\neg x$	$\neg x$	$[\neg x]_0^1$

From WMC to PBP

Example

- Indicator variable: x
- Parameter variables: p, q
- Weights: $w(p) = 0.2$, $w(q) = 0.8$, and $w(\neg p) = w(\neg q) = 1$

WMC Clause	In CNF	Pseudo-Boolean Function	
$\neg x \Rightarrow p$	$x \lor p$	$[\neg x]^{0.2}_{1}$	
$p \Rightarrow \neg x$	$\neg x \lor \neg p$		$[x]^{0.8}_{0.2}$
$x \Rightarrow q$	$\neg x \lor q$	$[x]^{0.8}_{1}$	
$q \Rightarrow x$	$x \lor \neg q$		
$\neg x$	$\neg x$	$[\neg x]^{1}_{0}$	$[\neg x]^{1}_{0}$

Outline

Correctness Conditions (1/2)

For each parameter variable $p \in X_P$,

- $w(\neg p) = 1$,
- and the set of clauses that mention p or $\neg p$ is

$$\left\{ p \vee \bigvee_{i=1}^{n} \neg l_i \right\} \cup \{ l_i \vee \neg p \mid i = 1, \ldots, n \}$$

for some non-empty family of indicator literals $(l_i)_{i=1}^{n}$.

In other words, p is defined to be equivalent to $\bigwedge_{i=1}^{n} l_i$.

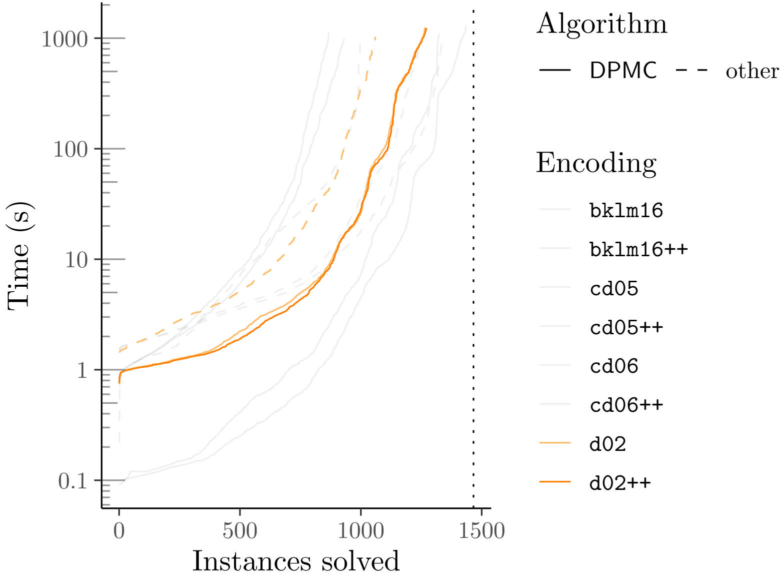
Correctness Conditions (2/2)

For each parameter variable $p \in X_P$,

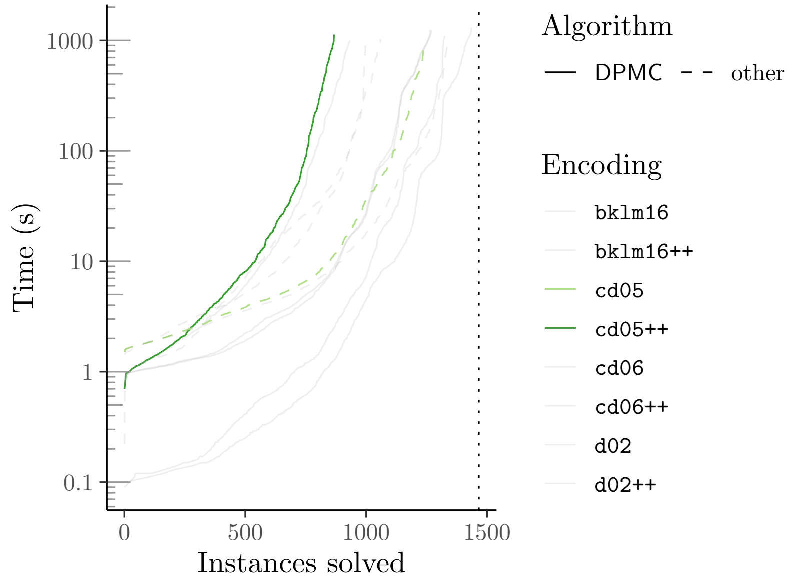
- $w(p) + w(\neg p) = 1$,
- each clause has at most one parameter variable,
- there are no negative parameter literals,
- if $\{p\} \in \phi$, then this is the only clause that mentions p,
- and for any two clauses of the form $\chi \Rightarrow p$ and $\psi \Rightarrow p$, $\chi \land \psi \equiv \bot$.

Outline

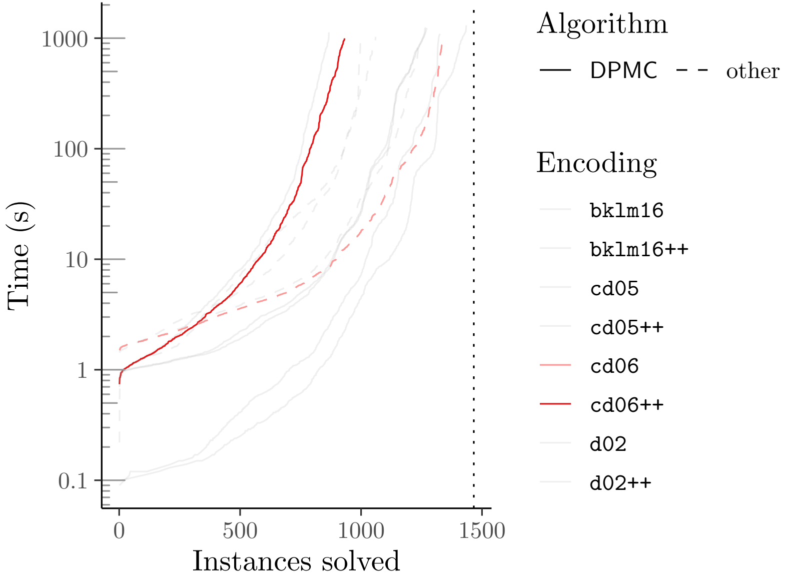
WMC/PBP Encodings for Bayesian Networks



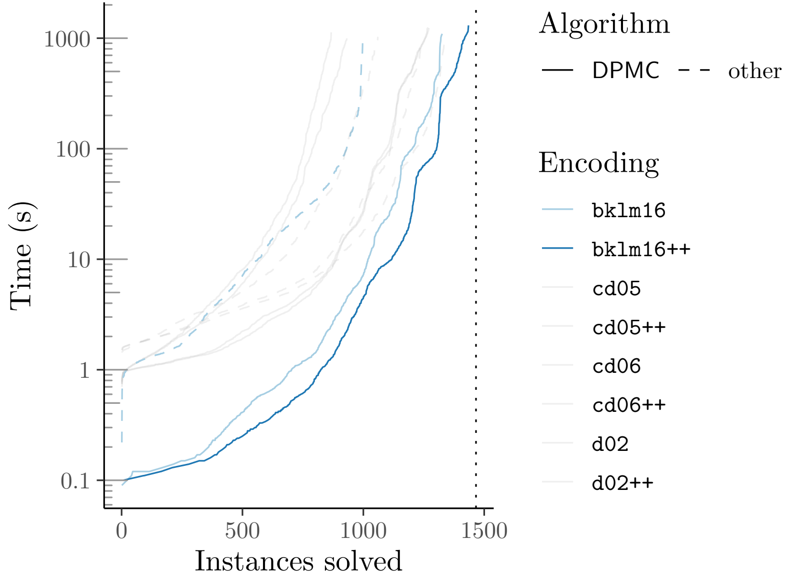
WMC/PBP Encodings for Bayesian Networks



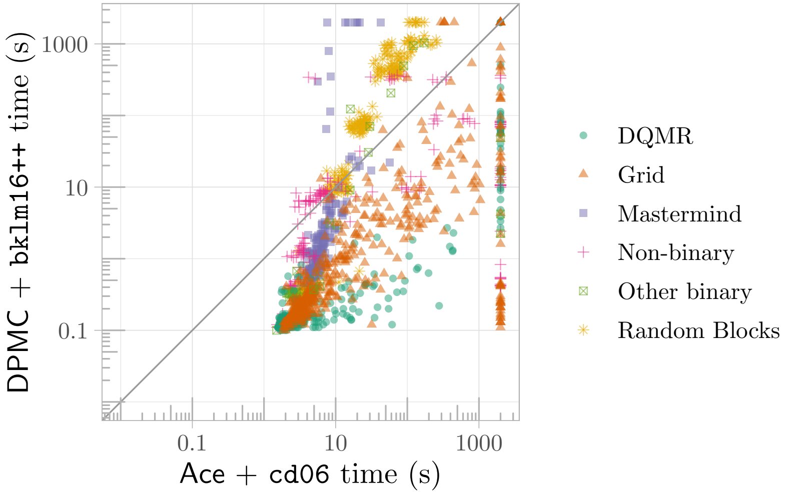
WMC/PBP Encodings for Bayesian Networks



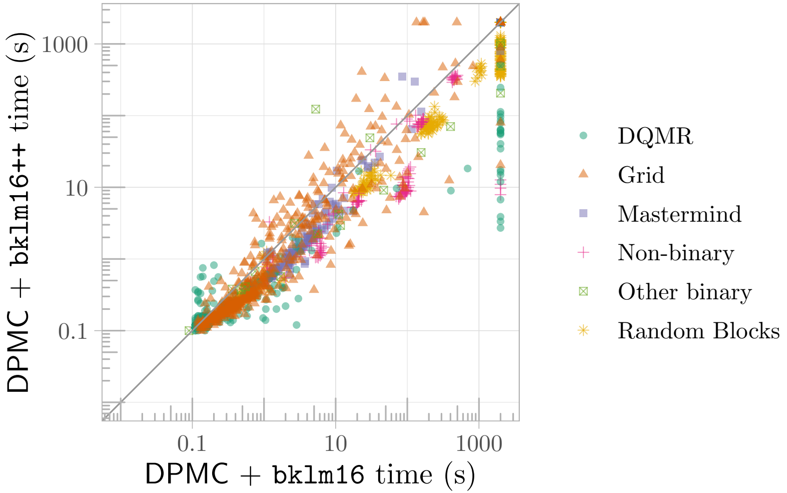
WMC/PBP Encodings for Bayesian Networks



Compared to the Previous State of the Art



The Best Encoding for DPMC: Before and After



Outline

Summary and Future Work

- PBP is a more expressive alternative to WMC that works with state-of-the-art WMC algorithms based on pseudo-Boolean function manipulation.
- Many WMC encodings can be efficiently transformed into PBP while removing unnecessary variables and clauses.
- The identified conditions for this transformation to work help explain how WMC encodings for Bayesian networks operate.
- Performance improvements depend on the encoding.
  - The very first encoding was virtually unaffected,
  - whereas the state-of-the-art encoding was significantly improved.
- Can the identified conditions be generalised further?
- Can the transformation be applied to WMC encodings for other application domains?